Individual Lab Report #11

Sumit Saxena

Team F: Rescue Rangers

Teammates:

Juncheng Zhang

Karthik Ramachandran

Xiaoyang Liu

April 18, 2017

Contents

1. Individual Progress

1.1. Overview

During the past two weeks, I worked on the following tasks:

1. GPS location estimation algorithm testing and improvements
2. Mount package drop mechanism on the drone
3. Migrate bright object detection algorithm to Python

1.2.1. GPS location estimation algorithm testing and improvements

To ensure that the GPS location estimation algorithm works well, I have been trying to test it as much as possible. We went for testing twice and once more for taking a final video for Progress Review, during the last two weeks.

Testing on April 11:

Procedure:

1. We placed the signatures at suitable locations and **recorded their actual locations using a third-party mobile app**
2. Conducted 4 flights over the signatures to collect data

Results:

Following are the GPS location estimation results from testing on one of the flights:

Signature 1: Sumit:
Actual location: 40.47241291035011, -79.96588750749822
Estimated location: 40.47242106066435, -79.9659809137096
Distance = 7m

Signature 2: Juncheng:
Actual location: 40.47227389648601, -79.96602052830156
Estimated location: 40.472267948343585, -79.96597227796094
Distance = 4m

Signature 3: Xiaoyang:
Actual location: 40.47236630696847, -79.96607542976733
Estimated location: 40.47233014762335, -79.96606054056858
Distance: 4m

Signature 4: Orange mattress:

Actual location: 40.47247221231505, -79.96602094739671
Estimated location: 40.47247898688512, -79.9659694889972
Distance: 4m

Signature 5: Karthik:
Actual location: 40.47251114625528, -79.96580259881910
Estimated location: 40.472513514087225, -79.96591057734177
Distance: 9m

Signature 6: Green mattress:
Actual location: 40.47234426256313, -79.96608817026015
Estimated location: 40.472429396369485, -79.96613408474377
Distance: 9m

Results just in terms of distances between actual and estimated locations have been misleading for us in the past. So, I plotted the locations on Google Earth and found some anomalies. The Google Earth plot is shown in Figure 1.

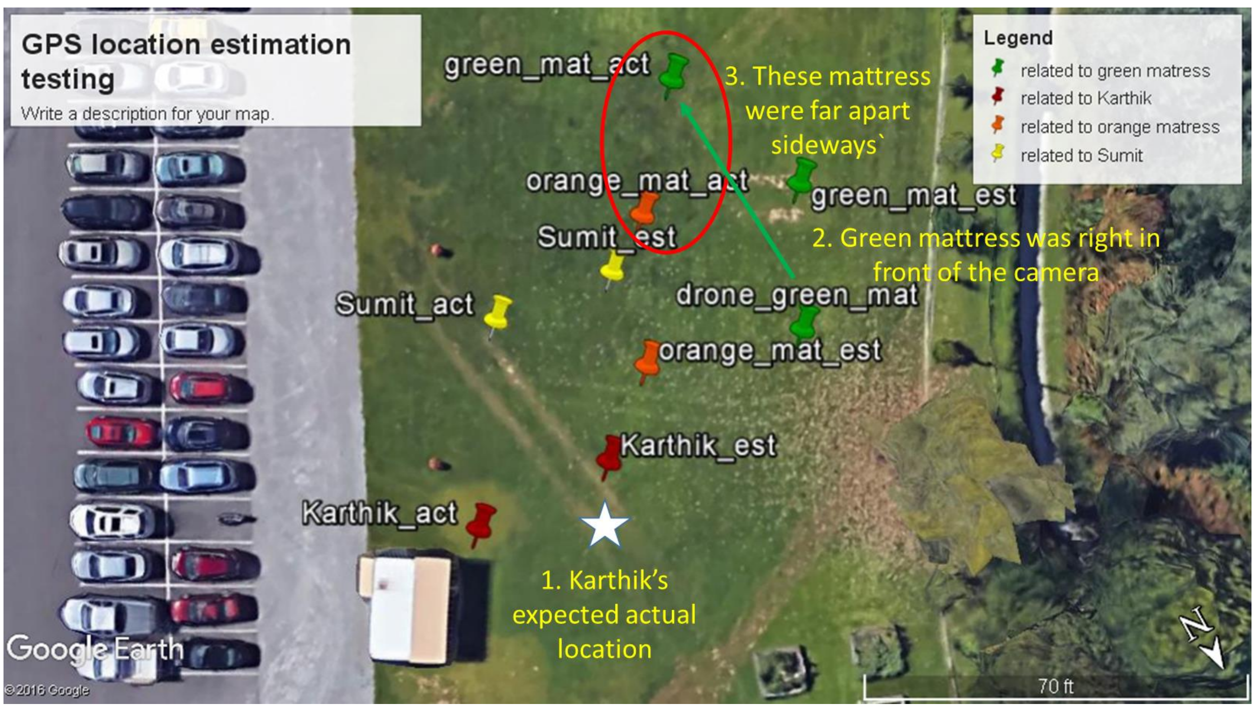

Figure 1: Google Earth plot showing locations of some of the signatures – both actual and estimated. Some of the anomalies have also been highlighted

Anomalies found:

1. **Karthik's actual location:** We knew from the setup that Karthik's actual location should have been somewhere near the location marked in Figure 1 as his expected actual location but the location we recorded using the mobile app is far away from that point.

2. **Green mattress's actual location:** We can see in Figure 1 that the green mattress's actual location and the drone location used to estimate its location are separated by a significant distance sideways. Figure 2 shows the frame from which the green mattress's location has been estimated. The green mattress was right in front of the camera. This means that either the green mattress's actual location or the drone's reported location is incorrect.
3. **Relative separation of orange and green mattresses:** We know from the setup that green mattress and orange mattress were placed to be in two separate sweeps of the flight and were separated by a significant distance sideways. But, their actual locations appear to be in a straight line in Figure 1.

Figure 2: Image frame used to estimate the green mattress's location.
This shows that the mattress was right in front of the camera when the shot was taken.

Conclusion:

We could not validate the working of the algorithm since the actual locations recorded for the signatures seemed to be flawed. We decided that we will record the actual signature locations using the drone's GPS during the next round of testing.

<u>Testing on April 16:</u>

Procedure:

We placed three human signatures in a straight row and recorded their GPS locations using the drone's GPS. Then, we conducted multiple flights over the signatures to collect data.

Results:

Testing the GPS location estimation algorithm directly on individual frames gave pretty good results. The results are shown below and also shown in Figure 3.

Xiaoyang:
Actual: 40.472373,-79.966046
Estimated: 40.472373842578534, -79.96607061851141
Distance: 2m

Juncheng:
Actual: 40.472428,-79.965918
Estimated: 40.472392177069764, -79.96598233658699
Distance: 6m

Sumit:
Actual: 40.472305,-79.966203
Estimated: 40.47229320023489, -79.96625477582533
Distance: 4m

Figure 3: Google Earth plot showing locations of all the signatures – both actual and estimated (second round of testing)

Further issues:
When we try to integrate signature detection algorithm with GPS location estimation algorithm, we get multiple locations for every signature because of every signature being detected in multiple discontinuous frames. Also, we get some false locations because of the false positives in detection.

Solution: Clustering:
We developed a clustering approach to cluster all the close-by GPS locations and return the centroid. We tested this as well the same day and got results similar to those shown above.
But, on further testing on some other videos, we found that the algorithm suffers sometimes when there are too many false positives.

1.2.2. Mount package drop mechanism on the drone

Since I could not get a good 3D print of the mounting plate (for mounting both the camera and package drop mechanism together) using the lab's 3D printer, I gave the model to NEA to have it printed there but we could not get the plate in time for testing during the weekend. I found an alternate way of mounting them both on a mounting plate, we had been using earlier for the 360fly camera, and I mounted them both on the drone. Figure 4 shows the mounting.

Figure 4: Camera and package drop mechanism mounted on the drone together

1.2.3. Migrate bright object detection code to Python

I also worked on migrating bright object detection code to Python and am facing some troubles in performing some morphological operations that I had used in MATLAB. There is a library 'Scikit-image' which has all the morphological operations required, though in slightly different form but I am facing some issues with the version of the library I am able to install. In the current state, the python code does not have very good performance and needs more work.

2. Challenges

1. Getting the ground truth GPS locations and correct visualizations:

It has been troublesome to analyze GPS location estimation. As discussed earlier in the report, some of the actual locations of the signatures we got from using a third party mobile app seemed to be incorrect. Also, it should be noted that the visualizations of these locations change a lot when we plot them on some other websites offering similar services. Thus, it has been difficult for us to validate the GPS location estimation algorithm well enough. Moving forward, we plan to use drone's GPS for everything.

2. Migrating bright object detection code to Python:

We are detecting bright objects using HSV thresholding followed by some morphological operations. This is working pretty well in MATLAB but I am not being able to install new versions of 'Scikit-image' python library which has all the required functions related to morphological image processing. Tuning the parameters of the limited functions I have available, has been difficult and has not yielded great results yet.

3. Teamwork

As a team, we did three rounds of outdoor flights and discussed issues with integration.
Work done by individual team members:

- Juncheng Zhang:
  - Migrated human detection algorithm including fusion to Python
- Sumit Saxena:
  - Tested and improved GPS location estimation algorithm
  - Mounted package drop mechanism on the drone
  - Worked on migrating bright object detection algorithm to Python
- Karthik Ramachandran:
  - Worked on data processing pipeline GUI
  - Included functionality to input rescue GPS location to the mobile app
  - Worked on autonomous package drop
  - Implemented clustering of output signature locations
- Xiaoyang Liu:
  - Worked on automating signature detection algorithm's inputs and outputs processing

4. Future plans

Following are the tasks I plan to work on until the SVE:

1. Further test and improve GPS location estimation algorithm – work on limiting the output to relevant locations
2. Test and improve bright object detection algorithm
3. Build a spare package drop mechanism
4. System integration